

# Framework de segmentación y análisis de imágenes mediante reconocimiento de texturas

Diego S. Comas<sup>1</sup>, Gustavo J. Meschino<sup>2</sup>, Virginia L. Ballarin<sup>1</sup>

<sup>1</sup> Laboratorio de Procesos y Medición de Señales, Facultad de Ingeniería, UNMdP

<sup>2</sup> Laboratorio de Bioingeniería, Facultad de Ingeniería, UNMdP

{diego.comas,gmeschin,vballari}@fi.mdp.edu.ar

**Resumen.** Cuando se combinan técnicas de extracción de características y reconocimiento de patrones para la segmentación de imágenes, la elección de las técnicas depende de la aplicación particular. El análisis de texturas constituye uno de los principales enfoques para la extracción de características. El conjunto de algoritmos desarrollados tanto para análisis de texturas como para reconocimiento de patrones es amplio. Cada técnica se basa en un determinado paradigma de análisis del problema y un conjunto de parámetros que la definen. La gran cantidad de parámetros involucrados hace a menudo difícil la tarea de selección y ajuste del algoritmo a utilizar. Frecuentemente es necesaria la combinación de técnicas para analizar la solución de una tarea de segmentación. Este trabajo propone la implementación de un *framework* que incluye un gran número de técnicas de análisis de texturas y reconocimiento de patrones facilitando la parametrización y combinación de técnicas para el análisis y segmentación de imágenes. Se analizan las características relacionadas con facilidad de utilización y tiempos de procesamiento.

**Palabras clave:** Procesamiento de Imágenes, Segmentación, Textura, Reconocimiento de Patrones, Framework.

## 1 Introducción

Múltiples problemáticas en el procesamiento digital de imágenes pueden analizarse con un enfoque de análisis de texturas [12]. El concepto de *textura* se puede definir como una disposición de puntos siguiendo un determinado patrón en forma regular y repetitiva a lo largo de los píxeles de la imagen que conforman el objeto. Es posible explicar el término *textura* como una repetición de elementos básicos denominados *texels*. Éstos pueden identificarse según diferentes aspectos, como por ejemplo: su patrón de intensidades, su rugosidad, sus componentes en frecuencia, sus rasgos morfológicos.

Las técnicas empleadas en el procesamiento de imágenes para la caracterización de texturas se basan en la asignación a cada región o píxel de una imagen un determinado conjunto de características que las describen [12]. La forma en que éstos se obtienen difiere de una técnica a otra. A este fin, se han desarrollado: técnicas de morfología matemática, técnicas propias de procesamiento de señales como Transformada de

Fourier y de Fourier-Mellin [6], técnicas probabilísticas, entre otras. La complejidad en la definición del concepto de textura hace difícil este proceso de caracterización. A menudo se combinan distintas técnicas de análisis buscando obtener un conjunto de características que represente adecuadamente un caso particular. El criterio para determinar los parámetros en cada técnica es relativamente complejo y debe estar acorde con la problemática a resolver. El tiempo de cómputo de los algoritmos se ha reducido con el avance de la velocidad de procesamiento, pero aún sigue siendo importante a la hora de seleccionar qué características obtener y con qué parámetros, debido a que en la mayoría de los casos se trabaja con un número grande de imágenes.

Luego de la extracción de las características, la información obtenida se utiliza para segmentar la imagen. Las técnicas de reconocimiento de patrones permitirán entonces agrupar los píxeles de la imagen. Existen también diversas técnicas para realizar esta tarea. Cuando no se dispone de información previa del agrupamiento esperado deben utilizarse métodos no supervisados, tales como: *K-means*, *Spectral Clustering*, mapas auto-organizados [8][9][11][15]. Cuando se tiene información previa sobre lo esperado (una imagen o una región previamente segmentada) se utilizan métodos supervisados, como por ejemplo redes neuronales multicapa o probabilísticas [8][14] o el algoritmo de K-vecinos más próximos [10], entre otros. El uso de cada técnica dependerá de la problemática particular. Es de suma importancia esta elección y habitualmente también se requiere la combinación de distintas técnicas. El conjunto de parámetros para configurar estos algoritmos y el tiempo de procesamiento no es tan crítico como en la extracción de características, pero la selección de las características a utilizar y la combinación de los algoritmos que definen el modelo planteado puede constituir una tarea no trivial, que por lo general requiere de gran número de pruebas.

Si bien hay software disponible para procesamiento de imágenes a nivel científico, como Ipp-plus [17], éstos no suelen incluir el reconocimiento de texturas. Posiblemente esto se debe a que se requiere tener conocimientos específicos de métodos de reconocimiento de patrones para trabajar con este enfoque. Otros programas tienen en cuenta el reconocimiento de textura, pero suelen estar enfocados a aplicaciones específicas [5][16]. El programa MatLab es una herramienta excelente, pero requiere conocer el lenguaje, y los scripts desarrollados para un determinado problema seguramente deberán ser re-escritos cuando se presenten nuevos tipos de imágenes. Hay infinidad de artículos que proponen el cálculo de diferentes características de texturas y distintos métodos de clasificación, pero no existe un software genérico configurable que los recopile, para poder efectuar pruebas en diferentes aplicaciones.

Se propone en este trabajo un *framework* que reúne un amplio conjunto de técnicas de análisis de texturas: matrices de coocurrencia, granulometría morfológica, transformadas de Fourier y de Fourier-Mellin, modelos fractales, y además permite utilizar diversas técnicas de reconocimiento de patrones.

Se incorporan técnicas para evaluar la clasificación o agrupamiento obtenido como: coeficientes de Tanimoto (uno por clase), coeficientes de exactitud, porcentaje de error, coeficiente de Dice y matrices de confusión.

El software desarrollado permite definir la totalidad de los parámetros que se utilizan en las técnicas incluidas, y además propone un conjunto de parámetros por defecto que facilitan el uso del mismo. En este trabajo se realiza un análisis de las técnicas implementadas y los tiempos de procesamiento involucrados.

## 2 Materiales y Métodos

### 2.1 Requerimientos del sistema y plataforma utilizada

En la propuesta inicial se consideró que el sistema debería constituir para el usuario una herramienta para el proceso de cálculo de características en imágenes y, de necesitarse, su posterior utilización en la segmentación. Los dos grandes módulos del sistema constituyen: la extracción de características y las técnicas de clasificación.

Se enuncian a continuación los principales requerimientos funcionales:

- El sistema debería permitir la completa parametrización de los algoritmos de procesamiento y reconocimiento de patrones. También debería incorporar parámetros por defecto en la totalidad de operaciones, facilitando la rápida implementación de pruebas sobre los algoritmos para usuarios no familiarizados.
- La *extracción de características* debería poder realizarse en una región de la imagen y procesar con los mismos parámetros un conjunto de imágenes numeradas en secuencia. Debería permitir realizar tanto un análisis píxel por píxel como por regiones, con la opción de realizar una caracterización de la imagen globalmente. El conjunto de características extraídas se almacenaría para un uso posterior de las técnicas de reconocimiento de patrones incluidas u otras técnicas no implementadas en el entorno desarrollado.
- Para el *reconocimiento de patrones*, el sistema debería permitir la selección de características a procesar y en qué región hacerlo, utilizando archivos de características del propio entorno u otros exportados desde otros sistemas.
- Para los algoritmos supervisados el sistema debería facilitar la selección de los datos de entrenamiento (regiones) y adicionalmente se debería construir una imagen pseudo-coloreada representando las clases resultantes.
- El sistema debería incorporar funciones de evaluación de las técnicas de agrupamiento.
- Sería esperable que el sistema presente información relacionada con el tiempo y esfuerzo de cálculo involucrados.

La totalidad de los algoritmos de procesamiento se implementó en MatLab® R2008a, herramienta con gran capacidad de prototipado en procesamiento de imágenes. En algoritmos de reconocimiento de patrones se utilizó la librería de redes neuronales y para la granulometría se utilizó la librería de morfología matemática, ambas incluidas en MatLab. El conjunto de ventanas y figuras fueron implementadas con la herramienta GUIDE. Esto permite, en primer lugar, una sencilla ampliación del sistema a futuro, debido a que el lenguaje en que está desarrollado es de uso común para la mayoría de los investigadores del área. En segundo lugar se encuentra la gran eficiencia de MatLab para operaciones relacionadas con el manejo de matrices, tipo de datos común en procesamiento de imágenes, reunidas con una variedad de controles para interfaces gráficas aptas para una aplicación con interfaz de usuario.

## 2.2 Técnicas Implementadas

### 2.2.1 Técnicas de reconocimiento de patrones

Las técnicas de reconocimiento de patrones buscan agrupar un conjunto de datos en una determinada cantidad de grupos (*clústeres*) mediante un mecanismo de decisión. El mecanismo dependerá de la técnica particular de la que se trate. El conjunto de datos estará formado por vectores de características representativas del problema a resolver (características de texturas, información de color, etc.). Cada dato será de la forma:  $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ . El conjunto de patrones será:  $\mathbf{X} = \{X_1, \dots, X_n\}$ , donde  $d$  es la dimensión del espacio de patrones y  $n$  es el número de patrones disponibles [9].

Los pasos de un algoritmo de *clustering* pueden enunciarse así: (1) Representación de los datos. Pueden incluirse etapas de selección y extracción de características; (2) Selección y aplicación de la técnica de reconocimiento de patrones (mecanismo de decisión); (3) Evaluación del desempeño, estimación del error e interpretación.

Las siguientes son algunas de las principales técnicas utilizadas:

*K-means*. En este algoritmo cada uno de los grupos (*clúster*) tendrá un centro en el espacio de patrones. Los datos son asignados a uno de los  $k$  clústeres según su distancia al centro sea la menor. El algoritmo se basa en la minimización de la suma de los cuadrados de las distancias de todos los puntos de un clúster al centro del mismo. El procedimiento se describe como sigue [9]: Se toman  $k$  centros de clúster iniciales  $\{Z_1(j), \dots, Z_k(j)\}$ , donde  $j$  representa la iteración. En la  $i$ -ésima iteración se distribuye el conjunto de datos  $\mathbf{X}$  entre los  $k$  centros. Se calculan los nuevos centros de clúster promediando los datos correspondientes a cada clúster. Si no varían entre dos iteraciones, el algoritmo se detiene.

Se implementó una optimización de este algoritmo que identifica datos repetidos, situación que generalmente se presenta en imágenes [1]. Esto reduce el tiempo de cómputo.

*Spectral Clustering*. En esta técnica se considera el conjunto de datos de entrada en relación a la distancia de cada uno de ellos con el resto. El uso de operaciones matriciales y de la teoría espectral de grafos permite obtener un conjunto de puntos que pueden ser agrupados fácilmente mediante técnicas no supervisadas como *K-means*. Los pasos del algoritmo son los siguientes [14]:

1. Generar la matriz afinidad  $\mathbf{A}$  de  $n \times n$ , definida como:

$$A_{ij} = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}} \text{ si } i \neq j \text{ y } A_{ij} = 0 \text{ si } i = j \quad (1)$$

2. Generar una matriz diagonal  $\mathbf{D}$ , cuyos elementos de la diagonal sean la suma de las filas de la matriz afinidad  $\mathbf{A}$ . Obtener  $\mathbf{L} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ .
3. Obtener los primeros  $k$  autovectores de  $\mathbf{L}$  (de los  $k$  autovalores más pequeños).
4. Generar una nueva matriz  $\mathbf{X}$ , de  $n \times k$ , donde cada columna de  $\mathbf{X}$  es uno de los  $k$  autovectores obtenidos en 3.

5. Normalizar la matriz  $\mathbf{X}$  obteniendo la matriz  $\mathbf{Y}$  según:

$$Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2} \quad (2)$$

6. Tratar cada fila de  $\mathbf{Y}$  como un patrón de un nuevo conjunto de patrones. Aplicar *K-means* a las filas de  $\mathbf{Y}$ .
7. Cada fila de  $\mathbf{Y}$  será asociada a uno de los  $k$  clústeres. Cada dato será asignado al clúster al que pertenece la fila correspondiente de la matriz  $\mathbf{Y}$ .

*K-vecinos más próximos.* En este método, se eligen  $N$  píxeles prototipo para cada una de las  $k$  clases a detectar. Cada uno de estos píxeles tendrá asociado un vector de características. La imagen completa es procesada píxel a píxel. Se extrae el conjunto de características y se obtiene el vecino más cercano a cada uno de los datos ingresados al algoritmo. Para esto se calcula la distancia euclidiana entre el dato correspondiente a la región y los patrones de los píxeles prototipos de cada clase. Se obtiene el más cercano en cada clase. La región se asigna a la clase más próxima entre los  $k$  vecinos extraídos [10].

*Mapas autoorganizados.* Son redes neuronales no supervisadas, basadas en observaciones sobre la organización topológica de la corteza cerebral. Están constituidos por dos capas, una de entrada y otra de salida [8]. La primera tiene la dimensión de los datos a procesar. La segunda es un arreglo de celdas (mapa) totalmente conectada con la capa de entrada. La cantidad de celdas del mapa es ajustable y dependerá de la aplicación. Los pesos del mapa se inicializan aleatoriamente o con valores del espacio muestral. Durante el entrenamiento, se obtiene la celda del mapa más cercana (distancia euclidiana) al dato y sus pesos y los de sus vecinas se actualizan para “acercarse” al dato. El porcentaje de cambio en el ajuste de los pesos disminuye a medida que el entrenamiento avanza. El objetivo de este tipo de redes es representar topográficamente los datos. Datos con características similares activarán las mismas regiones del mapa, por lo tanto esta técnica constituye una herramienta útil para observar la distribución de patrones [11].

*Redes neuronales multicapa.* Las redes neuronales multicapa están formadas, en tareas de reconocimiento de patrones, por un conjunto de entradas (una por cada dimensión de los datos), capas ocultas, y una capa de salida de  $k$  neuronas (una por clase). La red será *feed-forward* [8]. La selección de la cantidad de capas ocultas y el número de neuronas de cada capa oculta son heurísticos [14]. Los pesos de las conexiones sinápticas se inicializan en general en forma aleatoria. El aprendizaje constituye un proceso supervisado en que los datos de entrada y las salidas deseadas se presentan a la red. En cada iteración del proceso de aprendizaje, los parámetros libres de la red son ajustados llevando la respuesta de la red a la deseada. Una vez entrenada la red, en la etapa de consulta, se presentan nuevos vectores de características a clasificar. La neurona de la capa de salida con mayor nivel de activación indicará la clase a la que pertenece el dato consultado.

*Redes neuronales probabilísticas.* Las redes neuronales probabilísticas están constituidas por tres capas: una capa de entrada (una entrada para cada característica), una capa intermedia, y una capa de salida (con una neurona por clase) [8][14]. La capa de entrada envía información a la capa oculta, con función de activación gaussiana en el espacio de datos. Las neuronas de la capa de salida son lineales y su valor será una suma pesada de las salidas de la capa oculta. El proceso de aprendizaje consta de 2 etapas, una no supervisada y la otra similar a las redes multicapa [8]. La cantidad de neuronas de la capa escondida debe ser tal de cubrir el espacio de características. El diseño de la capa oculta puede efectuarse mediante el análisis del error o por métodos automáticos [17]. En la etapa de consulta, cada neurona de la capa de salida tendrá un determinado valor. La clase asignada al dato presente en la entrada será la correspondiente a la neurona con mayor nivel de salida.

### **2.2.2 Técnicas de extracción de características de texturas**

*Matrices de co-ocurrencia de niveles de gris.* El enfoque estadístico constituye una herramienta importante en el estudio de texturas. Brinda medidas de uniformidad o suavidad. La utilización de información espacial de la intensidad de los píxeles permite la caracterización de una textura. La matriz de coocurrencia se define como una matriz  $P(i,j)$  de frecuencias relativas en la que dos píxeles en la imagen separados una distancia  $d$  existen, uno con nivel  $i$  y otro con nivel  $j$ . Las matrices de coocurrencia son entonces dependientes de la distancia que separa a los píxeles considerados vecinos y la relación angular entre ellos. Se construyen considerando una ventana de análisis alrededor de un punto y contando la cantidad de veces en que un pixel con nivel  $i$  tiene un vecino, en una dirección y a una distancia determinada, con nivel de intensidad  $j$ . La matriz de frecuencias relativas denotará la probabilidad de tener en la región de análisis una transición de un nivel  $i$  a un nivel  $j$  en la dirección y distancia en que fue obtenida la matriz. Si la imagen tiene  $N$  niveles de gris, la dimensión de la matriz de coocurrencia será  $N \times N$ . Las matrices de coocurrencia serán simétricas respecto de la diagonal principal.

De la matriz de coocurrencia pueden obtenerse distintos valores que la describen, para formar el vector de características a utilizar en reconocimiento de patrones. Las principales características implementadas son: Contraste, Energía, Entropía, Media, Momento diferencial de orden  $k$ , Momento diferencial inverso de orden  $k$ , Directividad, Disimilaridad, Probabilidad Máxima [12].

*Granulometría morfológica en niveles de gris.* La Morfología Matemática realiza un análisis en las imágenes basado en operadores de la teoría de conjuntos, conceptos geométricos y topológicos. El método general se basa en recorrer la imagen y por cada píxel analizar un subconjunto de píxeles (región de análisis) definido por el elemento estructurante (EE) de la operación. En cada paso se realiza una función lógica determinada por la operación morfológica que se está utilizando [12].

El denominado Criterio de Tamaños, analiza como interactúa la imagen con EE crecientes utilizando operaciones básicas de la morfología como: erosión, dilatación, apertura y cierre. La Granulometría utiliza el Criterio de Tamaños con aperturas sucesivas. El EE que se utiliza es plano, es decir, define solamente el subconjunto de

píxeles. El método puede enunciarse como sigue:(1) Realizar la apertura morfológica con el EE;(2) Obtener el volumen (suma de los niveles de gris) de la imagen obtenida, para un entorno definido alrededor de cada pixel; (3) Dilatar el EE consigo mismo; (4) Iterar.

Del proceso se obtiene un conjunto de matrices que contienen los volúmenes calculados para cada iteración. Se define la Distribución Granulométrica de Tamaños:

$$G_n = 1 - \frac{\Omega(f \circ ng) - \Omega(f \circ n_0g)}{\Omega(f) - \Omega(f \circ n_0g)}, \quad (3)$$

donde:  $\Omega$ : Medida morfológica (volúmenes),  $f \circ ng$ :  $n$  aperturas sucesivas de  $f$  por  $g$  y  $n_0$ : Cantidad de aperturas realizadas.

Derivando la función distribución de la ecuación 3, se obtiene una función densidad de probabilidad. Obteniendo los momentos estadísticos de la misma (media, varianza y otros de mayor orden) se obtiene el conjunto de características [12].

*Transformada de Fourier.* Este enfoque utiliza la transformada discreta de Fourier. Propiedades como la rugosidad o suavidad de una textura son proporcionales al período espacial (tienen un determinado número de píxeles). Por lo tanto una textura gruesa (*suave*) deberá contener su energía en las bajas frecuencias espaciales. Al contrario, una textura fina (*rugosa*) tendrá valores altos del espectro concentrados en las altas frecuencias [12].

Para el análisis mediante métodos frecuenciales se utiliza el concepto de entropía. Este concepto representa la probabilidad de ocurrencia de distintas frecuencias y está ligada a la densidad espectral de potencia de la imagen.

Si una imagen es espacialmente periódica su espectro de potencia tiende a presentar picos en determinadas zonas en tanto que la concentración de valores del espectro en una cierta dirección expresan direccionalidad de la textura. Por esta razón, el espectro de frecuencia es dividido en regiones de interés (*bins*) cuando se realiza un análisis de texturas en el dominio de la frecuencia. Dichas regiones pueden ser radiales, en cuyo caso se tendrá en cuenta el contenido del espectro de frecuencia en regiones en forma de anillos alrededor de la baja frecuencia, o angulares. Los valores de entropía de las regiones serán los valores del vector de características [12].

*Transformada de Fourier-Mellin.* Las características obtenidas de las técnicas de análisis de texturas tradicionales varían cuando a la imagen se le aplica una transformación geométrica de rotación o escala. Ante un cambio de escala, por ejemplo, es preciso modificar el tamaño de la ventana de análisis de coocurrencia. La utilización de un conjunto de características invariantes ante transformaciones de este tipo, permite realizar una mejor caracterización de la textura en estudio y además permitirá la extensión de conocimientos sobre imágenes similares cambiadas de escala sin necesidad de procesamiento previo. De esta forma, texturas similares pero orientadas de diferentes formas o escaladas presentarán el mismo conjunto de características. A través de la transformada de Fourier-Mellin se pueden obtener un conjunto de características invariantes ante cambios de escala y rotación.

Para el cómputo en imágenes se realiza una discretización del dominio. Existen diferentes métodos de cálculo para esta transformada [6]. Para el análisis de texturas se utiliza, al igual que en el caso de Fourier, una vecindad alrededor de cada píxel.

*Modelos Fractales.* La caracterización de una imagen a través de la dimensión fractal es una técnica ampliamente usada [14]. Las dimensiones fractales brindan información sobre textura, típicamente su rugosidad. Una de las dimensiones fractales más utilizadas al caracterizar una textura es la relativa al exponente de Hurst.

Para la estimación del coeficiente de Hurst se utiliza el denominado rango de Quincux [14]. Se considera la ubicación del píxel como el centro de la región. Se obtienen los valores de la subregión para cada uno de las distintas distancias  $D(n)$  posibles al centro de la región. Para cada uno de estos conjuntos de niveles de gris se obtiene el rango  $R(n)$  del arreglo (diferencia entre el valor máximo y el valor mínimo). El coeficiente puede entonces estimarse como la pendiente resultante de una regresión lineal de los puntos  $[\log(D(n)), \log(R(n))]$ .

Un segundo método consiste en utilizar una vecindad fija. Cada píxel se recorre y se suman las diferencias entre los niveles de gris de un píxel con su vecino en la dirección de filas, y luego de columnas, ubicado a una determinada distancia. Esta distancia se incrementa iterativamente. El rango de cada iteración  $R(n)$ , se obtiene con el promedio de la suma de diferencias entre niveles [14].

### 2.2.3 Técnicas para evaluación de la clasificación

El uso de coeficientes que estiman la performance de la segmentación permite determinar la bondad del algoritmo de reconocimiento de patrones a nivel global e identificar particularidades de determinadas clases identificadas.

Las *matriz de confusión* presenta, para cada clase presente en el conjunto de resultados esperados, cuántos píxeles fueron clasificados en cada uno de los grupos. Es una matriz de  $N \times N$  clases (clase según los datos esperados y clases que fueron obtenidas). La diagonal de la matriz de confusión representa los datos agrupados correctamente. El resto de los valores representan errores y permiten obtener cuántos píxeles fueron erróneamente clasificados. Una valoración global del proceso de clasificación se puede obtener con un *porcentaje de error de clasificación*, obtenido como el cociente entre la suma de los elementos de la matriz de confusión (exceptuando a los de la diagonal) y la cantidad total de píxeles.

El *Coficiente de Tanimoto* para una determinada clase se define como:

$$T(A, B) = \frac{[A \cap B]}{[A] + [B] - [A \cap B]} \quad (4)$$

donde:  $[ ]$  denota la cantidad de elementos del conjunto,  $A$  es el conjunto de puntos de la clase obtenidos y  $B$  es el conjunto de puntos de la clase que se esperaba obtener.

Si  $T$  es 1, entonces la clasificación para esa clase fue perfecta; si  $T$  es 0, ningún punto de la clase fue clasificado correctamente.

El *Coficiente de Dice* se define a partir del coeficiente de Tanimoto como sigue:

$$D(A, B) = \frac{2 T(A, B)}{1 + T(A, B)} = \frac{2 [A \cap B]}{[A] + [B]} \quad (5)$$

donde:  $[ ]$  denota la cantidad de elementos del conjunto,  $A$  es el conjunto de puntos de la clase obtenidos y  $B$  es el conjunto de puntos de la clase que se esperaba obtener.

El *Coficiente de exactitud* es el cociente entre el número de elementos clasificados correctamente y el número de elementos esperados en esa clase:

$$E(A,B) = \frac{[A \cap B]}{[B]} \quad (6)$$

donde: [ ] denota la cantidad de elementos del conjunto, A es el conjunto de puntos de la clase obtenidos y B es el conjunto de puntos de la clase que se esperaba obtener.

### 2.3 Implementación

El diseño del *framework* consta de dos módulos: extracción de características y reconocimiento de patrones. Esto permite la separación de las etapas y la interactividad de los algoritmos implementados con otros externos; esto es, poder utilizar la extracción de características y procesar posteriormente en otras aplicaciones, o contrariamente, utilizar el módulo de reconocimiento de patrones con características obtenidos con otros sistemas. Las técnicas implementadas abarcan una gran cantidad de algoritmos utilizados tanto en análisis de texturas como en reconocimiento de patrones. La posibilidad de combinación de estas técnicas en una interfaz gráfica de usuario facilita la experimentación y uso.

La ventana de *Extracción de características* (Fig.1), permite la parametrización de las técnicas. Admite la selección de la imagen o imágenes a procesar y la región de procesamiento. Para permitir la repetitividad del conjunto de técnicas utilizadas por el usuario, el sistema permite almacenar la configuración de los algoritmos. Se muestra información relacionada con la cantidad de características que se extraerán e información de progreso en etapa de cálculo. Una vez obtenidos las características, pueden ser almacenados y visualizarse representaciones de los mismos en forma de imágenes (Fig. 2). Esto podría facilitar la selección y el ajuste de sus parámetros.

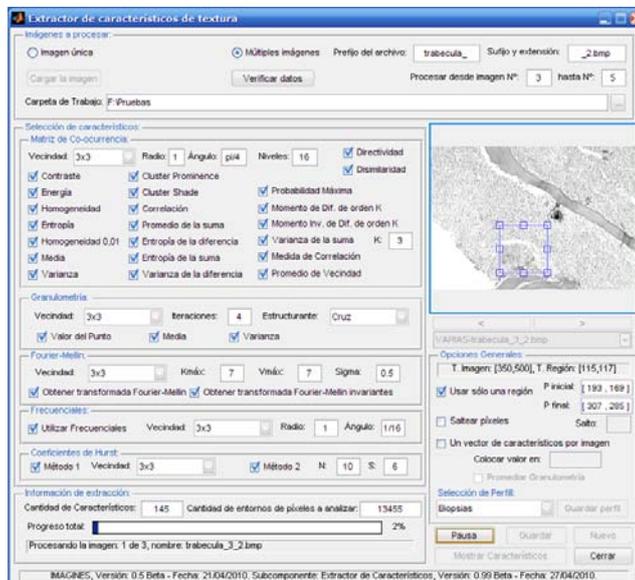
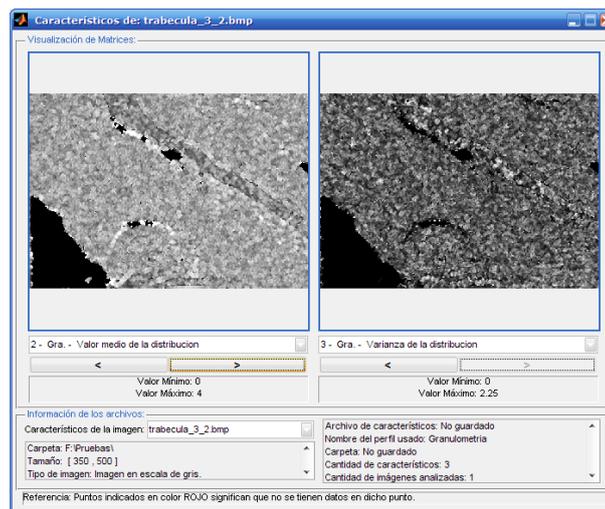


Fig. 1. Ventana de extracción de características.

El módulo de *Reconocimiento de patrones* fue a su vez dividido en dos tareas. La primera permite la selección del archivo de características a utilizar (generado por la aplicación o no), la selección de características a considerar, y la elección de la técnica de reconocimiento de patrones (Fig. 3). En cada una, pueden ser ajustados los parámetros que ésta requiera (cantidad de grupos  $k$ , datos de entrenamiento, arquitectura, etc.).

La segunda tarea es la aplicación de la técnica elegida. Luego de esto, el sistema permite generar una imagen pseudo-coloreada y calcula coeficientes de performance de las técnicas. Los parámetros configurados por el usuario en esta etapa pueden ser almacenados, como también las imágenes y datos resultantes.

El conjunto de parámetros que requiere cada técnica implementada es amplio. Cada aplicación particular tendrá un conjunto de características que describa el problema. Cada técnica de reconocimiento de patrones será óptima para un determinado conjunto de características. La selección de una u otra dependerá en gran medida de los datos disponibles sobre el problema a resolver (datos esperados, regularidad en la textura, forma de los objetos).



**Fig. 2.** Ventana de visualización de características.

El sistema facilita el uso de las técnicas proponiendo parámetros por defecto. Mediante pruebas y heurísticas conocidas, el usuario identifica el conjunto de parámetros óptimo para el caso particular. La funcionalidad de almacenar los ajustes realizados permite la repetitividad del procesamiento y la readaptación a nuevas problemáticas. La posibilidad de realizar un análisis sobre una secuencia de imágenes disminuirá el tiempo de intervención del usuario procesando la totalidad de las imágenes, una vez obtenido el conjunto de parámetros óptimos para una imagen representativa del problema.

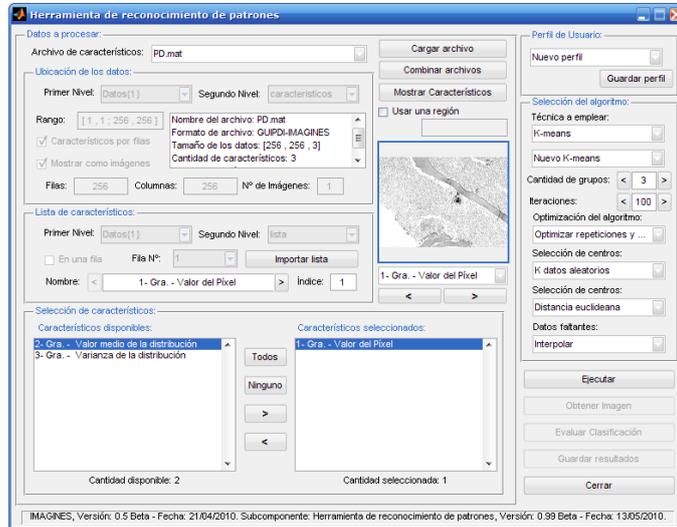


Fig. 3. Ventana de reconocimiento de patrones.

### 3 Resultados

El software desarrollado fue utilizado para segmentar imágenes de resonancia magnética de cerebro y de biopsias de médula ósea, entre otras, con el fin de evaluar la performance y analizar su adecuación a los requerimientos. Los algoritmos seleccionados fueron los considerados óptimos para los casos particulares analizados.

El principal aspecto a evaluar fue la adaptación del sistema a la configuración de los parámetros necesarios, tanto para la extracción de características como para los algoritmos de reconocimiento de patrones. Un segundo aspecto fue analizar el tiempo de procesamiento utilizado por la aplicación respecto del utilizado por las mismas técnicas sin utilizar interfaces gráficas de usuario.

El sistema permitió una sencilla configuración de los parámetros requeridos por el usuario (selección de imágenes, características, técnicas utilizadas).

Para la evaluación del tiempo de procesamiento, crítico en la etapa de extracción de características, se extrajeron características en 3 imágenes de médula ósea de 350 por 500 píxeles. Se analizaron 175000 píxeles utilizando todas las técnicas implementadas (145 características para cada píxel). Se utilizó una PC con procesador de 2 núcleos a una frecuencia de 2.2 GHz y una memoria RAM de 2GB. Los tiempos de procesamiento fueron del orden de 40 minutos. No hay modificación sustancial en el tiempo de procesamiento con o sin interfaz gráfica.

El uso de los perfiles de configuración facilitó la definición de los parámetros del sistema y permitió la repetitividad del procesamiento, aún fuera de los perfiles estándares incluidos en la aplicación.

El *framework* fue evaluado con imágenes de resonancia magnética de cerebro utilizando características granulométricas. Las características de las imágenes

multirresolución fueron combinados en un vector de 9 características (valor del punto, media y varianza de la distribución granulométrica de tamaños) utilizando la ventana de reconocimiento de patrones del sistema implementado. La segmentación se realizó mediante *K-means ponderado*. El tiempo de procesamiento promedio fue de 3.71 segundos para cada imagen de 256 por 256 píxeles.

## 4 Conclusiones

Se desarrolló un *framework* que permite el uso de un vasto y completo conjunto de herramientas para análisis de texturas y reconocimiento de patrones, con una interfaz gráfica optimizada para el procesamiento de imágenes. La combinación de este conjunto de técnicas se presenta óptima para cualquier problema de segmentación de imágenes mediante reconocimiento de texturas.

Se han abarcado los cálculos de características desde diversos enfoques, pudiendo calcularse todas en una sola ejecución, o seleccionarse solamente las que se desea obtener. Puede obtenerse un archivo con los valores obtenidos, que puede ser utilizado en otros sistemas, lo que le da a esta aplicación una gran potencialidad.

La configuración de cada una de las técnicas que se incluyen permite la adaptación de las mismas a cada problema particular. Se permite al usuario cambiar los parámetros que definen la ejecución de todos los algoritmos de cálculo, pero siempre se ofrecen parámetros por defecto, considerando que no siempre se conocerá cada método en profundidad. La interfaz gráfica implementada facilita la selección de los algoritmos y sus parámetros, brindando la posibilidad del almacenamiento de las configuraciones. Esto hace que el cálculo pueda repetirse en aplicaciones similares a las previas, permitiendo el reuso del conocimiento adquirido en experiencias anteriores exitosas.

El módulo de reconocimiento de patrones es adaptable para el procesamiento de datos extraídos mediante otras técnicas y permite a su vez la exportación de la totalidad de los resultados obtenidos. Se siguió el mismo criterio para las configuraciones de los algoritmos de reconocimiento en cuanto a su disponibilidad para el usuario y el almacenamiento de las mismas.

El módulo de evaluación permite que los resultados de las pruebas efectuadas puedan ser adecuadamente cuantificados, lo que facilita la tarea del reporte de errores cuando se requiere presentar resultados, requerimiento constante en la actividad académica.

La herramienta presentada permite desarrollar una gran cantidad de pruebas para lograr una segmentación deseada, sin tener que recurrir a la programación, permitiendo almacenar las configuraciones exitosas, lo que permite centrarse en los métodos probados y no estar pendiente de las dificultades típicas de la programación.

La implementación modular del sistema permitirá una sencilla ampliación a futuro de la aplicación. La incorporación de técnicas generales de procesamiento de imágenes complementará la aplicación y ampliará el conjunto de herramientas disponibles para la segmentación de imágenes. Asimismo, resulta perfectamente factible la incorporación de nuevas características y de nuevos métodos de reconocimiento de patrones, lo que hace de esta aplicación una propuesta general que será de gran interés en el contexto del procesamiento de imágenes.

## Referencias

1. Abras G. N., Ballarin V. L.: A Weighted K-means Algorithm applied to Brain Tissue Classification, JCS&T Vol. 5 No. 3 (2005)
2. Ballard D. H., Brown C. M.: Computer vision, Englewood Cliffs, N.J., Prentice-Hall. (1992). ISBN 0-1316-5316-4
3. Baxes G. A.: Digital image processing: principles and applications, New York, Wiley. (1994). ISBN 0-4710-0949-0
4. Bezdek J. C., Keller J., Krisnapuram R., Pal N. R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. (1999) . ISBN 978-0-7923-8521-9.
5. Carpenter A.E., Jones T.R., Lamprecht M.R., Clarke C., Kang I.H., Friman O., Guertin D.A., Chang J.H., Lindquist R.A., Moffat J., Golland P., Sabatini D.M.: CellProfiler: imageanalysis software foridentifying and quantifyingcellphenotypes, GenomeBiology, 7:R100 (2006)
6. Derrode S., Ghorbel F.: Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description, Computer Vision and Image Understanding, Vol. 83, No. 1 (2001), pp. 57-78
7. Gonzalez R. C., Woods R. E.: Digital image processing, Upper Saddle River, N.J., Prentice Hall (2002). ISBN 0-2011-8075-8
8. Haykin S.: Neural Networks. Macmillan College Publishing Company. Second Edition(1999)
9. Jain A.K., Murty M.N., Flynn P.J.: Data Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3 (1999)
10. Meschino G. J., Moler E. G.:Semiautomated Image Segmentation of Bone Marrow Biopsies by Texture Features and Mathematical Morphology, Analytical and Quantitative Cytology and Histology, Vol 26, No. 1 (2004), pp. 31-38
11. Meschino G., Passoni L., Scandurra A., Ballarin V.: Representación automática pseudo color de imágenes médicas mediante Mapas Autoorganizados. Título en resúmenes de 35ºJAIIO, Mendoza (2006)
12. Moler E. G., Ballarin V. L., Gonzalez M.:Características Estadísticos, Espectrales y Morfológicos para Clasificación de Texturas: Un Análisis Comparativo. VIII RPIC Reunión de Trabajo en Procesamiento de la Información y Control. Mar del Plata. Vol. 2 (1999) pp. 59-62
13. Ng. A. Y., Jordan M. I., Weiss Y.: On Spectral Clustering Analysis and an Algorithm, NIPS (2001)
14. Passoni L. I.: Characterizing Dynamic Speckle Time Series with the Hurst Coefficient Concept. Fractals ,No. 3(2004), pp. 319-330
15. von Luxburg U.: A Tutorial on Spectral Clustering, Statistics and Computing, 17 (4) (2007)
16. Szczypiński P.:Kociołek M., Materka A., Strzelecki M., Computer Program for Image Texture Analysis in PhD Students Laboratory, International Conference on Signals and Electronic Systems, Łódź-Poland(2001), pp.255-262
17. Wasserman P.D.: Advanced Methods in Neural Computing, New York, Van Nostrand Reinhold (1993), pp. 35-55
18. Media Cibernetics, <http://www.mediacy.com/>